Transactions Briefs

Analog CMOS Implementation of Cellular Neural Networks

Izzet Adil Baktir and Mehmet Ali Tan

Abstract— **The analog CMOS circuit realization of cellular neural networks with transconductance elements is presented. This realization can be easily adapted to various types of applications in image processing just by choosing the appropriate transconductance parameters according to the predetermined coefficients. The effectiveness of the designed circuits for connected component detection is shown by HSPICE simulations. For "fixed function" cellular neural network circuits the number of transistors are reduced further by using multi-input transconductance elements.**

I. INTRODUCTION

The term *artificial neural network* (ANN) has come to mean any computing architecture that consists of massively parallel interconnections of simple neural processors [1], [2]. Artificial neural networks try to mimic, at least partially, the structure and functions of the brain and the nervous system. The implementation artificial neural networks in VLSI takes advantage of the inherent parallelism to yield fast solutions [3]. In the VLSI implementation of neural networks, analog VLSI techniques are preferable because they lead to more compact and real-time realizations [4]–[7].

The cellular neural network (CNN) as proposed by Chua and Yang, is a special type of analog nonlinear processor array [8], [9]. Due to their continuous-time dynamics and parallel processing features, analog CNN circuits are very effective in real time image processing applications such as noise removal, edge detection and feature extraction [8]. The regularity, the parallelism and the local connectivity found in CNN circuit architecture make it suitable for VLSI implementations.

Motivated by the above facts we have implemented the CNN structure using analog CMOS circuits. One of our major goals is the design simplicity. To achieve this goal, the design is reduced to the design of few type of CMOS transconductance elements [10]. One can easily adapt this realization to various types of applications by just choosing the appropriate transconductance parameters according to the predetermined coupling coefficients between the neighboring cells. The coefficients may be either chosen according to a computer simulation or chosen based on the prominent kernels for image processing [11]. Another important motivation for using CMOS transconductance elements in the requirement of adaptability. In order to acheive programmable coupling coefficients, the transconductance parameters are adjustable with external voltage sources. In the implementation of "fixed function" CNN's that performs one or a related set of processing function using fixed coefficients, the number of transistors is reduced further by merging appropriate transistors in the multi-input transconductance subcircuit as explained in Section III.

Manuscript received July 8, 1992; revised October 20, 1992.

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey.

IEEE Log Number 9208101.

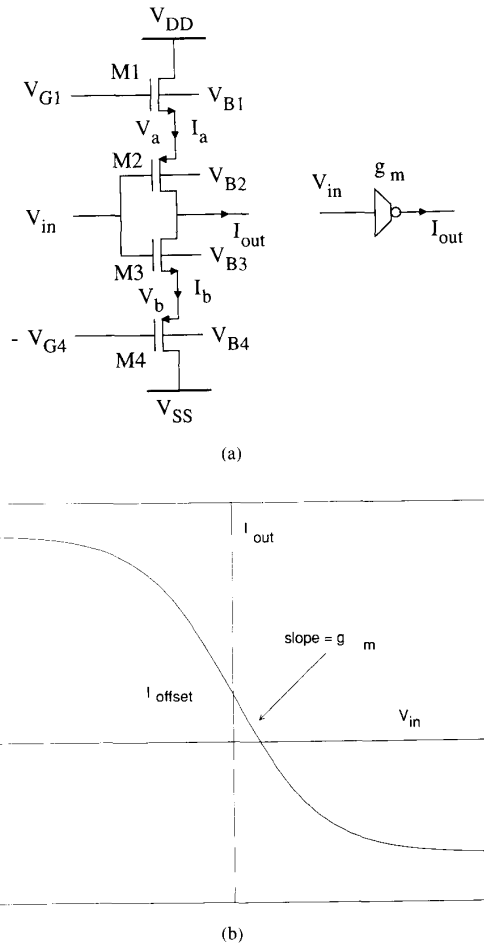


Fig. 1. (a) The transistor schematic and (b) the input–output characteristic of CMOS transconductance element.

II. ANALOG CMOS REALIZATION OF THE CELL CIRCUIT

Because of its simplicity, the CMOS transconductance element is chosen as the basic building block for the integrated realization of the CNN cell circuit [12]. The linear CMOS transconductance element (voltage-to-current transducer) resembles in most respects that of the CMOS inverter but without the matching problems between PMOS and NMOS transistors and with the additional advantage of tunability [12]. The transistor schematic and the input–output characteristic of this four-transistor transconductance element is shown in Fig. 1.

It can be easily proven that when all transistors operate in their saturation region [12], the output current $I_{\text{out}} = I_a - I_b$ equals

$$I_{\text{out}} = -g_m V_{\text{in}} + I_{\text{off}} \tag{1}$$

where the abbreviations

$$g_m = 2k_{\text{eff}}[V_{G1} + V_{G4} - (V_{Tn1} + V_{Tn3} + |V_{Tp2}| + |V_{Tp4}|)] \tag{2}$$

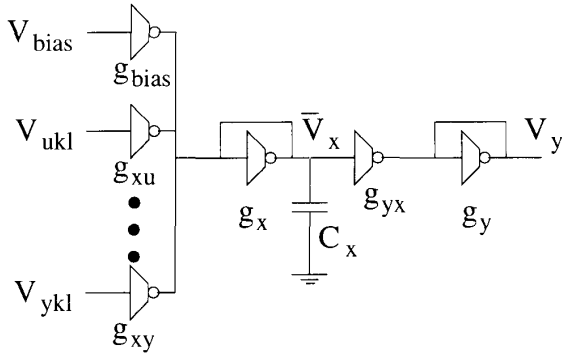


Fig. 2. The cell circuit realization with CMOS transducers.

$$I_{\text{off}} = \frac{g_m}{2}[(V_{Tn3} - V_{Tn1}) + (|V_{Tp4}| - |V_{Tp2}|) + (V_{G1} - V_{G4})] \tag{3}$$

are introduced for the transconductance parameter and the offset current, respectively.

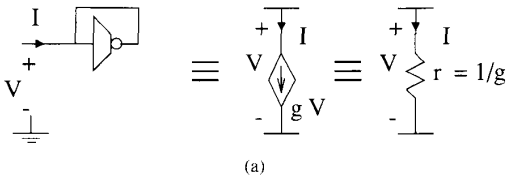
Although the offset current I_{off} is not equal to zero due to the body effect, it can be easily elminated by an appropriate setting of V_{B4} in an n-well process and V_{B1} in a p-well process [12].

2.1. Realization of a Cell with CMOS Transconductances

The circuit diagram of the integrated circuit realization of one CNN cell with CMOS transconductance element is shown in Fig. 2. It consists of the summation node v_x, where all the input currents and the bias current are summed, the state and output resistors, the input and output voltage-controlled current sources and a block realizing the sigmoid function.

The main problem in VLSI circuits is the implementation of the resistors that are not commonly used in standard CMOS technology. They usually occupy a large chip area which makes it impossible to implement networks with huge number of resistors. In order to eliminate this problem, we have implemented the cell circuit resistors R_x and R_y, by simply connecting the input of the transconductance to its output as shown in Fig. 3(a). The desired resistance values are achieved by choosing the gate area width-to-length ratio (W/L) appropriately after having set $V_{G1} = V_{DD} = 5$ V and $-V_{G4} = V_{SS} = -5$ V. Adjustable resistance can also be easily achieved by varying V_{G4} and V_{G1} externally. SPICE simulation results for different resistance values are shown in Fig. 3(b).

The sigmoid type nonlinear transfer characteristic needed at the output of the cell is performed with two transconductance elements. The second transconductance element whose input is connected to its output, acts like the resistor R_y and the output voltage v_y drives individual current sources whose outputs are coupled to the neighbors. Since input voltages v_u and the output voltages v_y are bounded by $\pm$ 1 V, the input control and the output feedback voltage-controlled current sources are obtained by using the transconductance elements $g_{xy}(i,j;k,l)$ and $g_{xu}(i,j;k,l)$ in their linear region.



(a)

Fig. 3. (a) The circuit diagram.

The desired coupling coefficient can be easily achieved by simply choosing the appropriate transconductance parameters. In order to perform one or a related set of processing functions using fixed coefficients, the transconductance parameter variation can be achieved by choosing the gate area W/L appropriately after having set $V_{G1} = V_{DD}$ = 5V and $-V_{G4} = V_{SS} = -5$ V. The negative (inhibitory) coupling coefficients can be obtained by inverting the positive (excitatory) input with a cascaded transconductance element pair as shown in Fig. 4.

The analog input data (initial state voltages and input voltages) and the output data (final output voltages) can be multiplexed row by row using two switches in each cell circuit [13]. Also, the synchronization of the transient continuous-time operation can be achieved by a start control signal [13].

In the designed circuit, the stability requirement of Chua [8]:

$$A(i,j;ij) > \frac{1}{R_x}$$

is achieved by choosing

$$g_{xy}(i,j;i,j) > g_x.$$

2.2. Programmable Coupling Coefficients

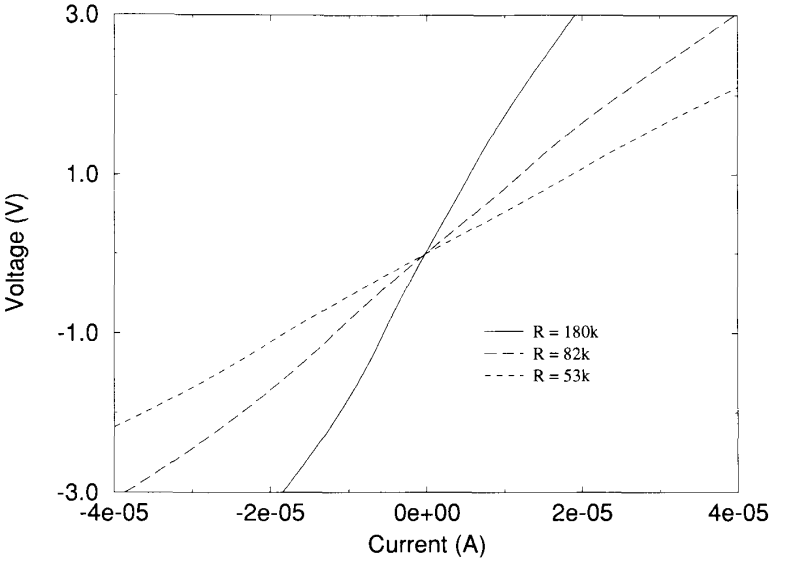
The requirement of adaptability is difficult to achieve in most neural network VLSI implementations [14], [15]. In the realization of the CNN structure with CMOS transconductance elements, we can achieve a programmable implementation by varying the transconductance parameters with external voltage sources connected to the gate voltages V_{G1} and V_{G4} of each cell defined in (2). HSPICE simulation results of the transconductance parameter variation for different gate voltages are shown in Fig. 5.

Nevertheless, one of the major concerns in VLSI programmable implementations of CNN is the wiring needed for changing the

$$g_{\text{eff}} = \frac{k}{2}\left[V_{DD} - V_{SS} - V_{Tn1} - |V_{Tp2}| - \sum_i w_i(|V_{Tp(2i+1)}| + V_{Tn(2i+2)})\right] \tag{5}$$

$$I_{\text{off}} = \frac{g_{\text{eff}}}{2}\left[V_{DD} + V_{SS} - V_{Tn1} + |V_{Tp2}| - \sum_i w_i(|V_{Tp(2i+1)}| - V_{Tn(2i+2)})\right] \tag{6}$$

$$\Delta = \frac{\left\{\sum_i w_i V_i^2 - \left[\sum_i w_i V_i\right]^2\right\}^2}{\left[-V_{DD} + V_{Tn1} + \sum_i w_i(V_i + |V_{Tpi}|)\right]^2 \left[[-V_{SS} - |V_{Tp2} + \sum_i w_i(V_i - V_{Tni})\right]^2}. \tag{7}$$



(b)

Fig. 3. (b) The characteristic of a CMOS resistor for different resistance values.

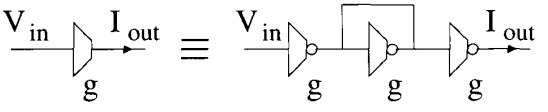


Fig. 4. Inhibitory coupling coefficients with CMOS transconductance elements.

template coefficients of each cell. Since all the cells of a CNN have the same coupling coefficients $g_{xy}(i,j;k,l)$ and $g_{xu}(i,j;k,l)$, the wiring problem can be largely eliminated by controlling the transconductance parameter variation of all cells with the same set of external gate voltages.

III. Reduced Hardware

In order to decrease the number of transistors in the realization of "fixed function" CNN structure, we propose a multi-input transconductance element which is shown in Fig. 6. The multi-input transconductance element is similar to m transconductance element proposed by Park *et al.*, except that upper NMOS and lower PMOS transistors are common [12].

In dc analysis of this new m-input voltage-controlled current source, assuming matching between NMOS transistor $M1$ and PMOS transistor $M2$ and using the standard square-law model for MOS transistors in their saturation region, the current I_{out} can be derived as

$$I_{\text{out}} = \left[-g_{\text{eff}} \sum_{i=1}^{m} w_i V_i + I_{\text{off}}\right][1-\Delta] \tag{4}$$

where g_{eff}, I_{off}, and Δ are as shown in (5)–(7) on the bottom of the previous page.

The conditions that should be satisfied in the derivative of (4) are

$$k = k_{n1} = k_{p2} = \frac{k_{n(2i+2)}}{w_i} = \frac{k_{p(2i+1)}}{w_i} \qquad i = 1,2,3,\cdots,m. \tag{8}$$

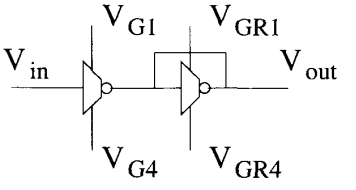


Fig. 5. (a) The circuit diagram.

and

$$\sum_{i=1}^{m} w_i = 1 \quad \text{and} \quad w_i > 0 \tag{9}$$

where k_{ni} and k_{pj} are the ith NMOS and jth PMOS transistor parameters, w_i's are the ratios between transistors and m is the number of input voltages.

The inhibitory coupling coefficients can be obtained by inverting the positive (excitatory) input with a cascaded transconductance element pair shown in Fig. 4. It can be easily shown that the maximum of the Δ term defined in (7) is less then 0.015, that is,

$$\max_{|V_i|\le 1, \sum w_i = 1} \text{percentage error} \le 1.5\%. \tag{10}$$

The derivations of the output current and maximum percentage error are given in Appendix A.

Since the maximum error is less the 1.5%, this multi-input VCCS can also be used in the implementation of fixed function CNN structures. SPICE simulation results of three-input VCCS

$$I_{\text{out}} = g_{\text{eff}}(w_1V_1 + w_2V_2 + w_3V_3) + I_{\text{off}} \tag{11}$$

for each input V_i $(i = 1,2,3)$ when $V_j = 0$ $(j \neq i)$ is shown in Fig. 7.

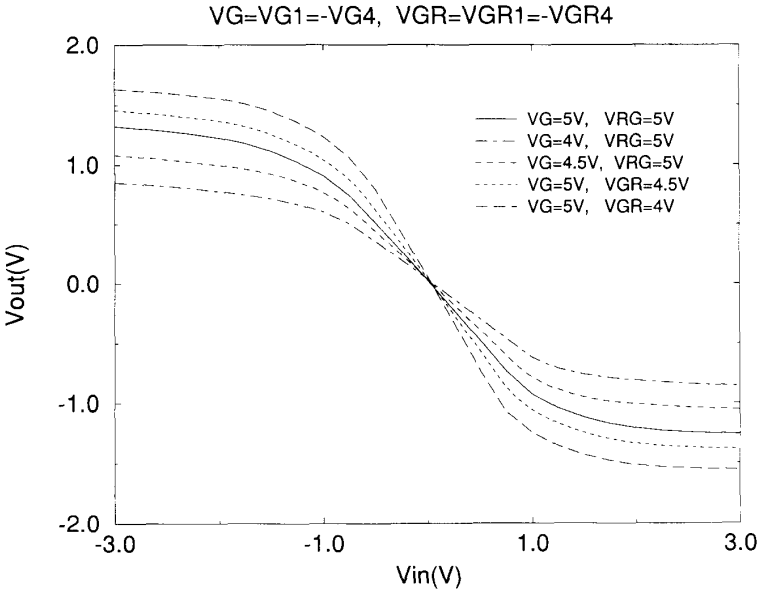


Fig. 5. (b) The simulation results of the transconductance parameter variation for different gate voltages.

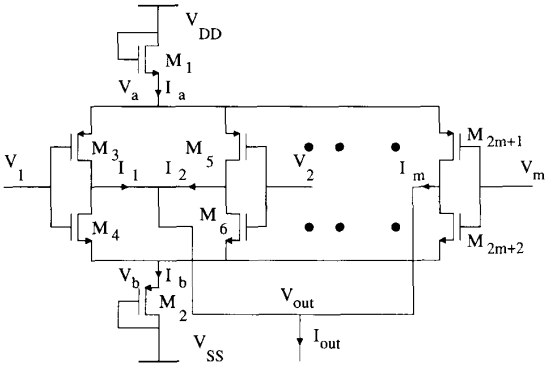


Fig. 6. M-input voltage-controlled current source.

IV. Simulation Results for Connected Component Detector

When the CMOS implementations described above are being used in an image processing application, the images are presented to the network as a set of initial input voltages to the state capacitors. The basic function of a CNN for image processing is to map an input image into a corresponding output image which are restricted with ± 1 V as pixel values [9], [16], [17]. How to choose the circuit parameters to achieve a desired image transformation is currently still an active research problem [18], [19].

In handwritten character recognition and many other feature extraction processes, the CNN circuits can be used as a connected component detector which detects the number of connected components of a vector in $\{+1, -1\}^N$ [13], [20].

To achieve the CNN cloning template for connected component detector given in [13], the parameters of our circuit are chosen as follows:

$$A = [g_x(i,j;i-1,j) \quad g_x(i,j;i,j) \quad g_x(i,j;i+1,j)] = g_x[1 \quad 2 \quad -1]$$
$$B = 0,$$
$$V_{\text{bias}} = 0$$
$$R_x = 100\,\text{K}\Omega$$
$$C = 4pF.$$

With the above circuit parameters, a 1×12 analog CMOS CNN circuit is simulated using the circuit simulator HSPICE. The set of initial conditions and the corresponding steady state values for the state voltages $v_{xi}(t)$ for $1 \leq i \leq 12$ is given in Table I.

To see the dynamic behavior of the circuit in more detail, the output transient characteristics of the state voltages are shown in Fig. 8. These transient characteristics show that the cell outputs reach their appropriate steady state values depending on both their neighbor cells and initial condition in less than 3.5 μs.

V. Conclusion

A new analog CMOS circuit implementation of a CNN has been presented. The design of CNN circuits is reduced to a transconductance element and it can be easily adapted to various types of applications by just tuning the appropriate transconductance elements according to the predetermined coupling coefficients between the neighboring cells in connected detector example.

The realization of CNN with CMOS transconductance elements can be either programmable or fixed function. In order to reduce the wiring problems in programmable CNN circuits, the desired coupling coefficients can be achieved by changing the transconductance parameters of each cell circuit with the same set of external voltage sources.

By using modified multi-input transconductance elements, the total number of transistors required for a cell circuit of connected component detector is reduced to 28. Since the number of transistors

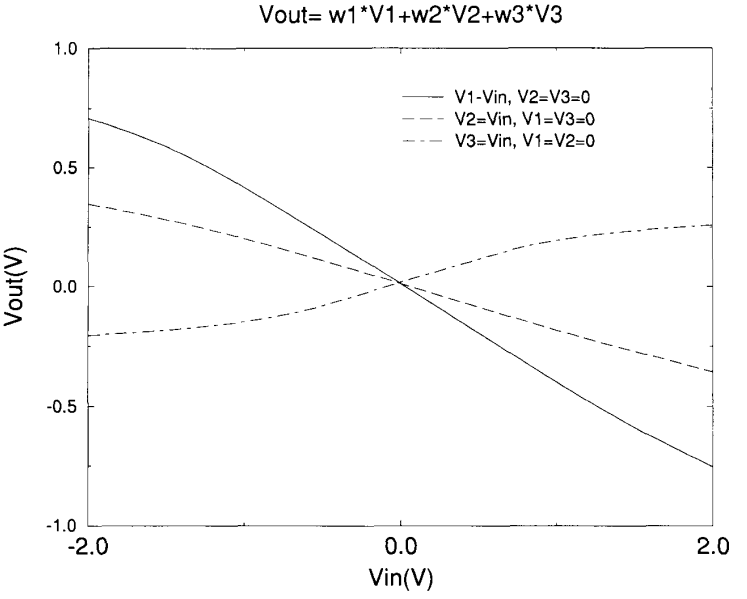


Fig. 7. Simulation results for $w_1 = -0.5, w_2 = -0.25, w_3 = 0.25$, and $g_{\text{eff}} = 1e-4\mho$. $I_{\text{out}} \approx g_{\text{eff}}(w_1V_1 + w_2V_2 + w_3V_3)$, for $|V_i| \leq 1, i = 1, 2, 3$.

TABLE I
HSPICE SIMULATION RESULTS FOR THREE DIFFERENT IINITIAL STATES

Simulation #1												
State	v_{x1}	v_{x2}	v_{x3}	v_{x4}	v_{x5}	v_{x6}	v_{x7}	v_{x8}	v_{x9}	v_{x10}	v_{x11}	v_{x12}
Initial	1.0	−1.0	−1.0	1.0	−1.0	−1.0	−1.0	−1.0	1.0	−1.0	−1.0	1.0
Final	1.4	1.4	1.4	1.4	1.4	2.0	−1.4	1.4	−1.4	1.4	−1.4	1.4

Simulation #2												
State	v_{x1}	v_{x2}	v_{x3}	v_{x4}	v_{x5}	v_{x6}	v_{x7}	v_{x8}	v_{x9}	v_{x10}	v_{x11}	v_{x12}
Initial	1.0	−1.0	−1.0	1.0	−1.0	1.0	−1.0	−1.0	1.0	1.0	−1.0	−1.0
Final	1.4	1.4	1.4	1.4	2.0	−1.4	1.4	−1.4	1.4	−1.4	1.4	−1.4

Simulation #3												
State	v_{x1}	v_{x2}	v_{x3}	v_{x4}	v_{x5}	v_{x6}	v_{x7}	v_{x8}	v_{x9}	v_{x10}	v_{x11}	v_{x12}
Initial	−1.0	−1.0	−1.0	1.0	1.0	−1.0	−1.0	1.0	1.0	−1.0	−1.0	−1.0
Final	1.4	1.4	1.4	1.4	2.0	−1.4	1.4	−1.4	1.4	−1.4	1.4	−1.4

needed is less than the previous approaches given in [13] and [14], with the proposed approach it is possible to increase the number of cells in the VLSI implementations of CNN's.

VI. APPENDIX A

For the m-input transconductance element shown in Fig. 6, let

$$k = k_{n1} = k_{p2} = \frac{k_{n(2i+1)}}{w_i} = \frac{k_{p(2i+1)}}{w_i} \quad i = 1, 2, 3, \cdots, m. \tag{11}$$

and

$$\sum_{i=1}^{m} w_i = 1 \quad \text{and} \quad w_i > 0 \tag{12}$$

where k_{ni} and k_{pj} are the ith NMOS and jth PMOS transistor parameters, w_i's are the ratios between transistors and m is the number of input voltages.

In dc analysis, using the standard square-law model for MOS transistors in their saturation region, the currents I_a and I_b, defined in Fig. 6, are easily derived as

$$I_a = \frac{k}{4}\left(\xi + \frac{\delta}{\xi}\right)^2 \tag{13}$$

and

$$I_b = \frac{k}{4}\left(\eta + \frac{\delta}{\eta}\right)^2 \tag{14}$$

where

$$\xi = -V_{DD} + V_{Tn1} + \sum_{i=1}^{m} w_i(V_i + |V_{Tp(2i+1)}|) \tag{15}$$

$$\eta = -V_{SS} - |V_{Tp2}| + \sum_{i=1}^{m} w_i(V_i - V_{Tn(2i+2)}) \tag{16}$$

and

$$\delta = \sum_{i=1}^{m} w_i V_i^2 - \left[\sum_{i=1}^{m} w_i V_i\right]^2. \tag{17}$$

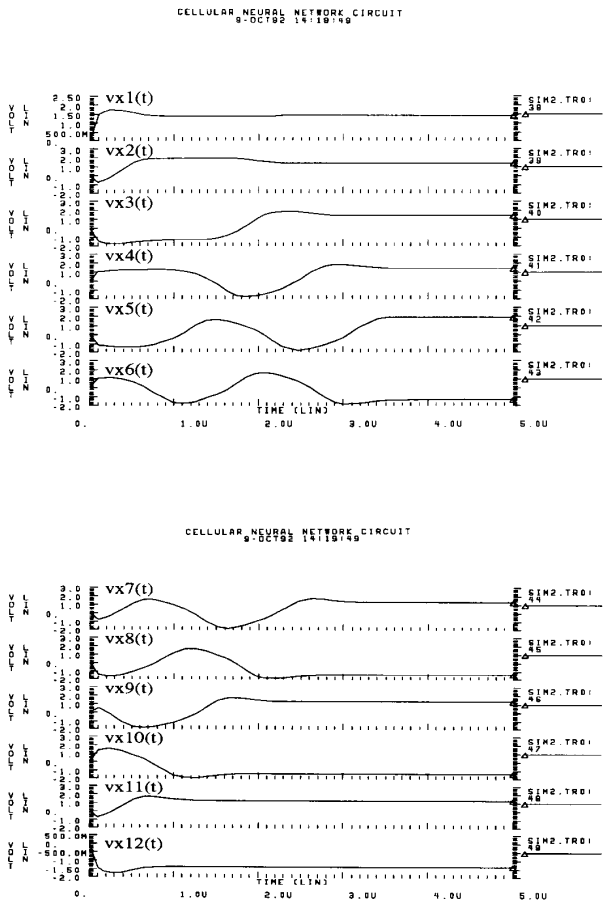


Fig. 8. Transient characteristics of the state voltages for second simulation.

Thus with (13) and (14), the output current $I_{\text{out}} = I_a - I_b$ equals

$$\begin{aligned} I_{\text{out}} &= \frac{k}{4}\left[\left(\xi + \frac{\delta}{\xi}\right)^2 - \left(\eta + \frac{\delta}{\eta}\right)^2\right] \\ &= \frac{k}{4}\left[\xi + \eta + \delta\left(\frac{1}{\xi} + \frac{1}{\eta}\right)\right]\left[\xi - \eta + \delta\left(\frac{1}{\xi} - \frac{1}{\eta}\right)\right] \\ &= \frac{k}{4}[\xi + \eta][\xi - \eta]\left[1 - \frac{\delta^2}{(\xi\eta)^2}\right] \end{aligned} \tag{18}$$

which implies

$$I_{\text{out}} = \left[-g_{\text{eff}} \sum_{i=1}^{m} w_i V_i + I_{\text{off}}\right][1 - \Delta] \tag{19}$$

where

$$\begin{aligned} g_{\text{eff}} &= \frac{k}{2}\Bigg[V_{DD} - V_{SS} - V_{Tn1} - |V_{Tp2}| \\ &\quad - \sum_i w_i(|V_{Tp(2i+1)}| + V_{Tn(2i+2)})\Bigg] \end{aligned}$$

$$\begin{aligned} I_{\text{off}} &= \frac{g_{\text{eff}}}{2}\Bigg[V_{DD} + V_{SS} - V_{Tn1} + |V_{Tp2}| \\ &\quad - \sum_i w_i(|V_{Tp(2i+1)}| - V_{Tn(2i+2)})\Bigg] \\ \Delta &= \frac{\delta^2}{(\xi\eta)^2}. \end{aligned} \tag{20}$$

The maximum percentage error can be written as

$$\begin{aligned} \text{max percentage error} &= \max \Delta \times 100 \\ &= \frac{\max |\delta|^2}{\min |\xi\eta|^2} \times 100. \end{aligned} \tag{21}$$

Since $|V_i| \leq 1$ and $\Sigma w_i = 1$

$$\begin{aligned} \max |\delta|^2 &= \max\left[\sum_{i=1}^{m} w_i V_i^2 - \left(\sum_{i=1}^{m} w_i V_i\right)^2\right]^2 \\ &= 1 \end{aligned} \tag{22}$$

and assuming $V_{T(n,p)i} \approx 1$ V, $V_{DD} = -V_{SS} = 5$ V

$$\begin{aligned} \min |\xi\eta|^2 &= min\left[\left(\sum_{i=1}^{m} w_i V_i\right)^2 - 9\right]^2 \\ &= 64. \end{aligned} \tag{23}$$

Inserting (22) and (23) into (21), the maximum percentage error is be obtained as

$$\begin{aligned} \text{max percentage error} &\leq \frac{1}{64} \times 100 \\ &\leq 1.5\%. \end{aligned} \tag{24}$$

REFERENCES

[1] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proc. IEEE,* vol. 78, pp. 1415–1442, Sept. 1990.

[2] Vemuri, Ed., *Artificial Neural Networks: Theoretical Concepts.* New York: IEEE Computer Society Press, 1986.

[3] U. Rueckert, K. Goser, U. Hilleringmann, and K. Schumacher, "VLSI technologies for artificial neural networks," *IEEE Micro,* pp. 28–44, Dec. 1989.

[4] C. Mead, *Analog VLSI and Neural Systems.* New York: Addison-Wesley, 1989.

[5] P. Mueller *et al.,* "Design and fabrication of VLSI components for a general purpose analog neural computer," in C. Mead and M. Ismail, Eds., *Analog VLSI Implementation of Neural Systems.* Norwell, MA: Kluwer, 1989, ch. 6, pp. 135–169.

[6] C. Mead and M. Ismail, Eds., *Analog VLSI Implementation of Neural Systems.* Norwell, MA: Kluwer, 1990.

[7] H. P. Graf and L. D. Jackel, "Analog electronic neural network circuits," *IEEE Circuits and Devices Magazine,* pp. 44–49, July 1989.

[8] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.,* vol. CAS-35, pp. 1257–1274, Oct. 1988.

[9] {—}{—}, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.,* vol. CAS-35, pp. 1275–1290, Oct. 1988.

[10] M. A. Tan, "Synthesis of artificial neural networks by transconductors only," *Analog Integrated Circuits and Signal Processing,* vol. 1, pp. 339–351, 1991.

[11] L. O. Chua and P. Thiran, "An analytical method for designing simple cellular neural networks," *IEEE Trans. Circuits Syst.,* vol. 38, pp. 1332–1341, Nov. 1991.

[12] C. S. Park and R. Schaumann, "A high frequency CMOS linear transconductance element," *IEEE Trans. Circuits Syst.*, vol. 33, pp. 1131–1138, Nov. 1986.
[13] J. M. Cruz and L. O. Chua, "A CNN chip for connected component detection," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 812–817, July 1991.
[14] L. Yang, O. Chua, and K. R. Krieg, "VLSI implementation of cellular neural networks," in *Proc. Int. Symp. on Circuits and Systems 1990*, pp. 2425–2427, May 1990.
[15] M. Verleysen and P. G. A. Jespers, "An analog implementation of Hopfield's neural network," *IEEE Micro*, pp. 46–55, Dec. 1989.
[16] T. Matsumoto, T. Yokohama, H. Suzuki, R. Furukawa, A. Oshimoto, T. Shimmi, Y. Matsushita, T. Seo, and L. O. Chua, "Several image processing examples by cellular neural networks," in *Proc. Int. Workshop on Cellular Neural Networks and Their Applications 1990*, pp. 100–106, Dec. 1990.
[17] G. Serker, "Small object counting with cellular neural networks," in *Proc. Int. Workshop on Cellular Neural Networks and Their Applications 1990*, pp. 139–148, Dec. 1990.
[18] F. Zou, S. Schwarz, and J. A. Nossek, "Cellular neural network design using a learning algorithm," in *Proc. Int. Workshop on Cellular Neural Networks and Their Applications 1990*, pp. 73–81, Dec. 1990.
[19] K. Slot, "Determination of cellular neural network parameters for feature detection of two dimensional images," in *Proc. Int. Workshop on Cellular Neural Networks and Their Applications 1990*, pp. 82–91, Dec. 1990.
[20] T. Matsumoto, L. O. Chua, and H. Suzuki, "CNN cloning template: Connected component detector," *IEEE Trans. Circuits Syst.*, vol. CAS-37, pp. 633–635, May 1990.

CMOS Implementation of an Analogically Programmable Cellular Neural Network

G. F. Dalla Betta, S. Graffi, Zs. M. Kovács, and G. Masetti

***Abstract*— The criteria to design the basic building blocks of an analogically programmable cellular neural network (CNN) in a 1.5-μm CMOS technology are reported. The simulated electrical performances of a 10×10 CMOS CNN, constituted of about 8000 MOS transistors, are presented and discussed. Finally, it is shown that the designed CNN can be successfully used to perform such useful functions as noise removal, edge detection, hole filling, shadow detection, and connected component recognition.**

I. Introduction

In recent years, a great deal of research work has been done on artificial neural networks, as they can potentially be used for several practical applications such as speech and image recognition, whereas normal processors require a considerable amount of processing time. In particular, neural digital systems, either in parallel or serial form, neural analog electronic networks and neural optical systems have been suggested [1], [2].

Very recently, a new and alternative approach to the classic artificial neural network topologies, called cellular neural networks (CNN's), was introduced by Leon Chua *et al.* [3], [4]. Immediately after their introduction, CNN's raised great interest as an alternative to conventional computers for image processing and pattern recognition [5]–[8], [10]–[18].

Manuscript received July 24, 1992; revised November 6, 1992.
The authors are with the Department of Electronics, University of Bologna, 40136 Bologna, Italy.
IEEE Log Number 9208102.

CNN's exhibit several interesting features and properties; among these the more important are:

a. each cell (or processing unit) is locally connected only to its neighbors;
b. the processed signals are analog in nature and the continuous-time operation allows the computational speed to depend only on the time-constant of the underlying dynamic system;
c. the architecture of both the processing unit and of the interconnection elements between cells is very simple and space invariant; this renders the CNN's particularly suited for an implementation in a VLSI technology, like CMOS.

Up to now, only a few proposals of practical CNN implementations have been published [5]–[8]. In particular, in [6] a CMOS design in a current-mode approach of a 10×10 simple low-pass filter for noise removal has been proposed, while in [7] a 6×6 connected-component detector (CCD) has been designed, fabricated, and experimentally tested. As a matter of fact, the work of [7] represents the only CNN chip fabricated up to now. All of these designs refer to given established applications for which the weights of the interconnections between elementary cells (the so-called template weights) are fixed.

However, in practice, the possibility to manage a neural network to be used for different applications in different time periods of the signal processing procedure is particularly interesting, as one can use the same part of the chip to perform several circuit functions. In this context, a CMOS approach for the design of a reconfigurable cellular network in which the template coefficients can be digitally varied was proposed in [5]. Unfortunately, however, even if this approach is very interesting from a circuit point of view, the control circuitry required to digitally select the template coefficients values leads to a very complex final architecture.

In this work we will present the design, in a 1.5-μm CMOS technology, of an analogically programmable CNN architecture with low-power dissipation which can be adopted for several applications in image processing. Additionally, the analogically programming feature can also be used to test the robustness of CNN performances against undesired variations in the values of the template coefficients.

In Section 2.1 the design of the elementary processing unit and of the linear voltage-dependent current sources that implement the feedforward template coefficients is presented. Section 2.2 is devoted to the design of the CMOS multiplier which enables to analogically program the "lateral" template coefficients of the feedback operator. Section 2.3 briefly discusses the differences between the ideal model and the CMOS implementation. Section III reports the electrical performances of a 10×10 CMOS CNN, constituted of about 8000 MOS transistors and fully simulated at the device level, which can simply be programmed by varying an external control voltage. By doing so, the designed CNN can perform such useful functions as noise removal, edge detection, hole filling, shadow detection and connected component recognition. Finally, some conclusions are drawn in Section IV.

II. Design

2.1 Cell Core and Control Operator

The basic building blocks of the CNN have been developed with reference to a well-assessed 1.5-μm n-well CMOS technology with an oxide thickness of 250Å. A circuit supply voltage of ± 5 V was considered.